Task Name: Qt/QML UI Design for CAN open Configuration on Raspberry Pi

Brief: This project involves designing a user-friendly **Qt/QML-based UI** on **Raspberry Pi** for configuring CAN open communication. The interface will allow users to read sensor data, manage Electronic Data Sheet (EDS) files, and configure CAN open devices efficiently. The goal is to provide an interactive UI for real-time CAN open configuration.

Infrastructure Requirements:

1. Qt/QML Installation on Raspberry Pi

* Install the Qt framework and QML runtime environment.
* Configure the Raspberry Pi OS to support Qt-based UI applications.

2. Integration with PCAN Basic API

* Ensure the UI can communicate with CAN open devices via PCAN Basic API.
* Implement backend logic for CAN open message handling.

3.Real-Time Data Display & Interaction

* Develop an I nterface that dynamically updates with real-time CAN data.

4.CAN open Protocol Support

* Implement CAN open-specific message handling for device communication.
* Support standard CAN open functions such as NMT (Network Management), PDO (Process Data Object), and SDO (Service Data Object) transactions.

Configuration and Housekeeping Requirements:

1. UI Layout & Design

* Develop a structured UI with distinct sections for CAN device management,configuration settings, and real-time data monitoring.

2. CAN Interface Initialization

* Enable and configure the CAN interface.
* Ensure proper baud rate for seamless communication.

3. CAN open Parameter Configuration

* Implement user-friendly UI for modifying and saving CAN open settings.
* Allow users to set and update CAN parameters via UI elements such as Node ID.

**5.** EDS File Management

* Enable loading, parsing, and validation of Electronic Data Sheet (EDS) files.

4.Logging & Debugging Tools

* Maintain a log of configuration changes and communication errors.
* Provide error messages and status updates for troubleshooting.

Functional Use Case Requirements:

### 1. **System Initialization**

* **Manual Action Required:** User initializes the CAN communication.
* **System Response:** Displays **system status** and available **CAN open nodes**.

### **2. Sensor Data Monitoring**

* **Recognized User Actions:**
  + "Start Monitoring" → Begins **real-time sensor data acquisition.**
  + "Stop Monitoring" → Stops **data acquisition.**
* **System Response:**
  + Displays **live sensor values** from connected CAN open nodes.
  + Alerts the user if **communication errors** occur.

### 3. **CAN open Configuration UI**

* **Recognized User Actions:**
  + "Load Configuration" → Load an **existing CAN open configuration**.
  + "Modify Configuration" → Edit **configuration parameters**.
  + "Save Configuration" → Save **updated parameters.**
  + "Apply Configuration" → Write **settings to the CAN open device.**
* **System Response:**
  + Confirms **successful loading, saving, and application** of settings

### 4.**EDS File Handling**

* **Recognized User Actions:**
  + "Load EDS File" → Opens an **Electronic Data Sheet (EDS) file.**
* **System Response:**
  + Provides **validation feedback** and confirms successful loading.

### **5. Configuration Writing to CAN open Device**

* **Recognized User Actions:**
  + "Write Configuration" → Writes **modified parameters** to the CAN device.
* **System Response:**
  + Confirms **successful data transmission** or alerts in case of failure.